프로토콜 리버스 엔지니어링을 위한 객관적 평가 및 검증 방법에 관한 연구

구영훈, 심규석, 이민섭, 박지태, 김명섭
고려대학교
{gyh0808, kusuk007, chenlima2, pjj5846, tmskim}@korea.ac.kr

A Study of Evaluation and Validation Method for Protocol Reverse Engineering

Young-Hoon Goo, Kyu-Seok Shim, Min-Seob Lee, Jee-Tae Park, Myung-Sup Kim
Korea Univ.

요 약

인터넷 사용의 대중화와 네트워크 발전은 대용량의 다양한 트래픽을 발생시키고 있다. 이러한 환경 하에 발생하는 프로토콜 중 다수는 비공개 프로토콜이다. 효과적인 네트워크 관리를 위하여 비공개 프로토콜의 사양을 추출하는 프로토콜 리버스 엔지니어링 기술은 반드시 선행되어야 한다. 다양한 프로토콜 리버스 엔지니어링 방법론의 연구에 비해 추출된 결과를 검증하는 방법은 매우 더딘 것이 현실이다. 본 논문에서는 프로토콜 리버스 엔지니어링의 결과를 검증하는 방법론을 제시한다. 다각적인 평가 및 검증 요소를 정의함으로써 다양한 리버스 엔지니어링 방법론의 성능을 평가하고 검증함은 물론 여러 다른 방법론과의 성능 비교 역시 가능하게 한다.

Ⅰ. 서론

대용량의 인터넷 회선이 보편화되고 인터넷 서비스를 사용하는 디바이스가 급격히 증가함에 따라 인터넷 트래픽이 급증하고 다양해졌다. 이러한 환경 하에 발생하는 프로토콜 중 다수는 알려지지 않은 비공개 프로토콜이다. 따라서 비공개 프로토콜의 사양을 추출하는 프로토콜 리버스 엔지니어링의 중요성이 커지고 있다. 이러한 프로토콜 리버스 엔지니어링은 네트워크 관리와 제어 측면에서 많은 이점을 가진다.

여러 다양한 방법론을 적용한 프로토콜 리버스 엔지니어링에 관한 연구는 활발히 진행되는 반면, 제안하는 방법론과 이를 이용한 분석 결과를 객관적으로 평가하는 기준에 대한 연구는 아직 미비한 실정이다.

객관적인 평가 기준의 부재는 제안한 방법론의 정확한 성능 평가와 다양한 방법론 간의 성능 비교를 어렵게 한다. 따라서 본 논문에서는 프로토콜 리버스 엔지니어링의 객관적인 검증과 평가 방법을 제안한다.

2장에서는 문제를 정의하고, 3장에서는 다각적인 평가 및 검증 요소를 정의하며 4장에서 결론을 제시한다.

Ⅱ. 문제 정의

프로토콜 리버스 엔지니어링의 주된 성능 평가 사항으로는 프로토콜의 구문이 올바른 메시지 포맷을 나타내는가가 가장 중요하다. 프로토콜 구문이 정확하다는 전제하에 이를 토대로 한 의미 및 유한 상태 머신의 올바른 추론이 가능하기 때문이다.

메시지 포맷 추출 알고리즘의 효과를 측정하고 결과를 비교하려면 이를 위한 성능 평가 방안과 성능 평가 지표를 정의해야 한다. 기존의 많은 연구들에서는 저마다의 성능 평가 방안을 통해 방법론을 평가하고 있지만, 적용하는 체계가 통일되지 않아 서로 다른 방법론의 성능을 비교하기가 쉽지 않다. 또한, 기존의 연구에서 사용하는 성능 평가 지표만으로는 결과에 대한 명확한 검증을 하기에 부족하다.

Case	방법론	성능 평가 방안
1	Polyglot, Pext, FieldHunter, Veritas	정성적 평가 방안
2	Discoverer, Netzob ,AutoReEngine	Correctness, Conciseness, Coverage
	Prospex, Biprominer, ReverX	Soundness, Completeness, Precision, Recall
3	WasP, AutoFormat, Dispatcher, Tupni	자체 평가 방안

표 1. 기존 연구 방법론의 성능 평가 방안 분류

표 1은 기존 연구의 성능 평가 방안을 각 Case별로 분류한 표이다. Case1과 같이 정성적인 평가만을 사용하는 연구도 있으며, Case2와 같이 유사한 성능 평가 지표를 사용하지만 서로 다른 이름을 사용하는 연구들도 있다. 그리고 Case3과 같이 자체 성능 평가 지표를 사용하여 평가하거나 해당 방법에만 사용할 수 있도록 국한되어 있는 성능 평가 지표를 사용하는 연구도 존재한다. 예를 들어, AutoFormat의 finest grain fields, Hierarchical fields, parallel fields들의 분류 및 Tupni의 Loop 탐지 횟수를 토대로 한 성능 평가 지표의 경우 범용적으로 사용할 수 없다.

Case2의 경우 True Positive, False Positive, False Negative를 토대로 한 성능 평가 지표를 사용하거나 이와 유사하지만 명확한 메시지 포맷의 검증을 위해 개선된 Correctness, Conciseness, Coverage를 성능 평가 지표로 사용하지만 명확한 검증을 하기에는 부족하다. 예를 들어 Coverage는 추출된 메시지 포맷으로 얼마나 많은 실제 메시지를 분석할 수 있는지를 측정하는 메시지 단위의 평가 지표이지만, 추출된 메시지 포맷으로 얼마나 많은 실제 정답지 메시지 포맷을 분석할 수 있는지의 정답지 메시지 포맷 단위의 평가는 존재하지 않으며, 추출된 메시지 포맷의 필드 포맷 세분화 정도, 추출된 메시지 포맷이 얼마나 강건하게 실제 메시지들의 유형을 분류하였는지에 대한 평가 요소는 미비하다.

한편, 추출된 메시지 포맷을 검증하기 위한 정답지 메시지 포맷의 기준이 모호하여 서로 다른 방법론의 비교 및 성능 평가 결과에 대한 신뢰성이

이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No.2015R1D1A3A01018057)과 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. 2017-0-00513, Security Analytics 기반의 이기종 보안솔루션 위협 분석 및 대응 기술 개발)

부족하다. 예를 들어, HTTP 프로토콜의 경우, 크게는 HTTP 요청 메시지 포맷, HTTP 응답 메시지 포맷 2가지로 분류할 수 있으나 HTTP 요청 메시지의 경우 Method 필드가 어떤 값을 사용하느냐에 따라 더 세부적으로 분류할 수도 있고, Method 필드가 같은 값을 사용하는 두 메시지의 경우라도 구성하는 Header Name 필드들의 시퀀스에 따라 서로 다른 메시지 포맷으로 볼 수 있다.

Ⅲ. 객관적 평가 및 검증 방법

3.1 트래픽 선정 조건

서로 다른 방법론에 대한 객관적인 평가를 위해서는 트래픽의 선정 조건이 중요하다. 우선 입력 트래픽은 동일하여야 하며, 여러 호스트 및 다양한 환경에서 수집하는 것이 옳다. 또한 올바른 검증을 위해서 입력 트래픽은 결함이 있는 플로우가 없어야 한다. 결함이 있는 플로우는 4가지로 분류된다. 플로우의 발생 중간 시점부터 수집된 플로우, 캡쳐 디바이스의 속도 한계로 인해 캡쳐되지 못한 패킷이 있는 플로우, 프로토콜 구문이 실제 명세에 위반되는 패킷을 포함한 플로우, 패킷 역전 및 재전송에 의한 패킷 중복이 있는 플로우가 그것이다.

3.2 정답지 생성 방안

프로토콜 리버스 엔지니어링의 성능 평가는 비공개 프로토콜을 입력으로 하였을 때 나온 결과에 대한 정답을 알 수 없으므로, 현재 알려져 있는 상용 프로토콜을 입력으로 하여 나온 결과를 상용 프로토콜의 명세와 비교하여 얼마나 일치하는지를 판단하는 방법만이 유일무이하다.

또한 각 프로토콜별로 구조가 상이하기에 각 실제 명세를 통해 프로토콜별로 정답지를 만드는 방법을 정의하여야 한다. 기존의 많은 연구에서는 검증 대상 프로토콜로 텍스트 기반 프로토콜로는 HTTP를 바이너리 기반 프로토콜로는 DNS 프로토콜을 사용하고 있다. HTTP 프로토콜, DNS 프로토콜을 예시로 텍스트 및 바이너리 기반 상용 프로토콜의 정답지를 생성하는 방안을 설명한다.

HTTP 프로토콜의 경우 URL과 Value 및 Body는 상대적으로 값이 너무 가변적이거나 주요 필드가 아니므로 HTTP 프로토콜에서의 f(True Field)는 Request Message의 경우 Method, Version, Header Name에 해당하는 값들로 설정하고 Response Message의 경우 Version, Status code, Phrase, Header Name에 해당하는 값들로 설정하고 m(True Message) 메시지를 구성하는 f의 시퀀스로 설정한다.

DNS 프로토콜의 경우 Query Name, Domain name, Time to live, Resource data length, Resource data 필드는 상대적으로 너무 가변적이거나 주요 필드가 아니므로 DNS 프로토콜에서의 f(True Field)는 Query Message의 경우 Header와 Query Type, Query Class에 해당하는 값들로 설정하고 Response Message의 경우 Header와 Query Type, Query Class, Domain class로 설정하고 이에 해당하는 값들로 설정한다. 이 때, DNS는 바이너리 기반 프로토콜이므로 각 바이트 위치에서 나오는 값들을 f로 설정하고 이에 대한 중복을 제거한 값을 TF(True Field Format)으로 설정한다. 예를 들어, Header의 첫 번째 byte에서 나오는 값들의 중복을 제거한 값 : TF_{1B} = {0x01, 0x0d, ... }, Header의 두 번째 byte에서 나오는 값들의 중복을 제거한 값 : TF_{2B} = {0x01, 0x0c, ... }, ...과 같이 TF가 설정된다. 따라서 m(True Message)은 메시지를 구성하는 f(각 바이트 위치에서의 값)의 인접한 연속을 말한다.

3.3 메시지 포맷에 대한 성능 평가 지표

• Correctness

$Correctness_{EMi}$는 개별 Extracted Message Format을 평가하기 위한 값으로 True Message Format을 기준으로 평가한 값이다. 해당 Extracted Message Format으로 전체 True Message Format들 중 얼마나 많은 True Message Format을 매칭할 수 있는지를 의미하며 해당 Extracted Message Format에 매칭되는 True Message Format의 개수 / 전체 True Message Format의 개수로 $Correctness_{EMi}=n(TM|TM\in EMi) / n(TM)$으로 나타낼 수 있다. 전체 Extracted Message Format을 평가하기 위한 값인 $Correctness_{Total}$은 전체 Extracted Message Format으로 매칭되는 True Message Format의 개수 / 전체 True Message Format의 개수로 $Correctness_{EMi}$의 합집합을 의미한다. 따라서 $Correctness_{Total} = \cup\{Correctness_{EMi}\} = \cup\{n(TM|TM\in EMi) / n(TM)\}$으로 나타낼 수 있다.

• Coverage

$Coverage_{EMi}$는 개별 Extracted Message Format을 평가하기 위한 값으로 실제 메시지를 기준으로 평가한 값이다. 해당 Extracted Message Format으로 전체 메시지들 중 얼마나 많은 메시지를 분석할 수 있는지를, 즉 매칭할 수 있는지를 의미하며 해당 Extracted Message Format에 매칭되는 메시지의 개수 / 전체 메시지의 개수로 $Coverage_{EMi} = n(Message|Message\in EMi) / n(Message)$로 나타낼 수 있다. 전체 Extracted Message Format을 평가하기 위한 값인 $Coverage_{Total}$은 전체 Extracted Message Format으로 전체 Extracted Message Format으로 전체 메시지들 중 얼마나 많은 메시지를 분석할 수 있는지를, 즉 매칭할 수 있는지를 의미하며 모든 Extracted Message Format에 매칭되는 메시지의 개수 / 전체 메시지의 개수로 $Coverage_{Total} = \sum\{Coverage_{EMi}\} = \sum\{n(Message|Message\in_{EMi}) / n(Message)\}$로 나타낼 수 있다.

• Detail

$Detail_{EMi}$는 개별 Extracted Message Format을 평가하기 위한 값으로 결과로 추출한 Message Format이 얼마나 필드가 세분화되어 있는지를 평가하기 위한 값이다. $Detail_{EMi} = n(f|f\in(EMi)) \times n(TM|TM\in EMi) / n(f|f\in(TM\in EMi))$의 수식으로 구할 수 있다. 전체 Extracted Message Format을 평가하기 위한 값인 $Detail_{Total}$은 전체 Message Format들이 얼마나 필드가 세분화되어 있는지를 평가하기 위한 값으로 모든 $Detail_{EMi}$에 대한 평균과 같다. 따라서 $Detail_{Total} = \sum Detail_{EMi} / n(EM)$으로 나타낼 수 있다.

• Compression

Compression은 추출한 Message Format이 입력 메시지들에 대하여 충분히 압축하여 클러스터링한 결과인지에 대한 평가 요소로 $Compression = n(EM)/n(Message)$로 나타낼 수 있다.

Ⅳ. 결론 및 향후 연구

본 논문에서는 프로토콜 리버스 엔지니어링의 필요성을 제시하고 기존의 여러 방법론들의 무분별한 평가 및 검증 방법에 대해 문제점을 제시하였다. 객관적인 평가 및 검증 기준의 부재는 프로토콜 리버스 엔지니어링 연구의 발전을 더디게 만든다. 검증 대상 트래픽 선정 조건 및 정답지 생성 방안과 메시지 포맷에 대한 성능 평가 지표를 정의하였다. 본 논문에서 제안한 객관적 요소는 다양한 방법론의 성능을 정확히 측정할 뿐만 아니라, 다른 방법론과의 비교를 가능하게 한다.

참 고 문 헌

[1] John Narayan, Sandeep K. Shukla, T. Charles Clancy, “A Survey of Automatic Protocol Reverse Engineering Tools”, Journal ACM Computing Survey, Vol. 48, Issue. 3, No. 40, 2016.